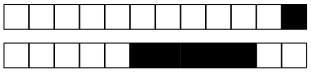


Entraînement - Training

INSTRUCTION : *English version below*

En haut de chaque page se trouvent 3 nombres, par exemple +1/3/58+. Vous ***devez*** *vérifier que, sur chacune des pages de votre sujet, le* ***premier*** *de ces 3 nombres est le même (dans cet exemple, il s'agit donc du 1). Ce nombre identifie votre copie. Les deux autres nombres ne sont pas importants.*

Détacher la dernière feuille et répondre dessus. Ne pas rendre les pages contenant les questions, vous ne devez rendre ***que la dernière feuille****. Chaque question est sur 1 point, aucun point ne sera attribué aux questions contenant une mauvaise réponse.*

Les questions faisant apparaître le symbole ♣ peuvent présenter une ou plusieurs bonnes réponses qui doivent toutes être cochées. Les autres ont une unique bonne réponse.

At the top of each page are written 3 numbers, +1/3/58+. You ***must*** *check that, on each page you have, the* ***first*** *number is the same (in this case, it would be the number 1). This number is the id of your subject. The two other numbers are not important.*

Answer only on the last page. Keep the other pages containing the questions, you just have to return ***the last page****. Each right answer gives you 1 point. For any wrong answer, the mark of the question is 0.*

If there is a question with a symbol ♣, there may be one or more right answer. All of them must be checked. Any other question has only one right answer.

Question 1

6 jobs must be executed on a machine M_1 then on a machine M_2, in that order. At the end of the execution of a job on M_1, it can be executed on M_2 if the machine is available, otherwise, we wait for the availability to start the job on M_2. The durations of the jobs on the two machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5	j_6
M_1	50	180	190	80	210	30
M_2	60	240	230	90	140	100

Check the order that could be returned by the Johnson algorithm.

☐ 1 125436	☐ 3 513624	☐ 5 463215	☐ 7 241635	☐ 9 614235
☐ 2 623154	☐ 4 135642	☐ 6 642513	☐ 8 462153	☐ 10 125634

Question 2

6 jobs must be executed on a machine M_1 then on a machine M_2 and finally on a machine M_3, in that order. At the end of the execution of a job on M_1, it can be executed on M_2 if the machine is available, otherwise, we wait for the availability to start the job on M_2. Similarly for M_3. The durations of the jobs on the three machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5	j_6
M_1	160	150	190	230	110	60
M_2	60	30	60	20	20	50
M_3	190	220	160	130	180	200

Parmi les ordres suivants, lequel correspond à un résultat que pourrait renvoyer l'algorithme de Johnson adapté au cas de 3 machines ? Si cet algorithme n'est pas adapté au cas de 3 machines, cochez la case "Err".
Check the order that could be returned by the Johnson algorithm, if this algorithm is suitable for this case. Otherwise, check "Err".

1 Err	3 451362	5 563412	7 146253	9 312546
2 162534	4 341562	6 531642	8 321546	10 652134

Question 3

8 jobs must be executed on a machine M_1 then on a machine M_2, in **any** order. A job can be executed on one machine at a time, and a machine can execute only one job at the same time. The durations of the jobs on the two machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5	j_6	j_7	j_8
M_1	1630	260	190	240	160	220	260	260
M_2	760	10	60	20	170	210	220	30

What is the minimum time we need to execute all the jobs on the machines ?

1 3310	3 2680	5 1680	7 2390	9 1480
2 2010	4 3220	6 2870	8 1880	10 2410

Question 4

6 jobs must be executed on a machine M_1 then on a machine M_2, in that order. At the end of the execution of a job on M_1, it can be executed on M_2 if the machine is available, otherwise, we wait for the availability to start the job on M_2. The durations of the jobs on the two machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5	j_6
M_1	180	50	90	240	140	220
M_2	60	160	160	40	170	210

Check the order that could be returned by the Johnson algorithm.

1 543126	3 235614	5 261354	7 362145	9 162534
2 624153	4 432165	6 453126	8 326514	10 246531

Question 5

8 jobs must be executed on a machine M_1 then on a machine M_2 and finally on a machine M_3, in that order. At the end of the execution of a job on M_1, it can be executed on M_2 if the machine is available, otherwise, we wait for the availability to start the job on M_2. Similarly for M_3. The durations of the jobs on the three machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5	j_6	j_7	j_8
M_1	190	130	270	320	310	90	190	310
M_2	60	30	30	40	30	30	80	70
M_3	200	80	250	230	250	140	120	290

Parmi les ordres suivants, lequel correspond à un résultat que pourrait renvoyer l'algorithme de Johnson adapté au cas de 3 machines ? Si cet algorithme n'est pas adapté au cas de 3 machines, cochez la case "Err".
Check the order that could be returned by the Johnson algorithm, if this algorithm is suitable for this case. Otherwise, check "Err".

1 Err	3 82546371	5 51627438	7 62184735	9 56738412
2 75614328	4 87215364	6 53427186	8 63154827	10 61835472

Question 6

6 jobs must be executed on a machine M_1 then on a machine M_2, in **any** order. A job can be executed on one machine at a time, and a machine can execute only one job at the same time. The durations of the jobs on the two machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5	j_6
M_1	200	220	40	160	200	60
M_2	200	20	140	60	160	10

What is the minimum time we need to execute all the jobs on the machines ?

1 480	3 400	5 760	7 310	9 880
2 630	4 700	6 390	8 590	10 740

Question 7

6 jobs must be executed on a machine M_1 then on a machine M_2, in that order. At the end of the execution of a job on M_1, it can be executed on M_2 if the machine is available, otherwise, we wait for the availability to start the job on M_2. The durations of the jobs on the two machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5	j_6
M_1	170	10	50	20	20	180
M_2	110	220	90	240	60	160

Check the order that could be returned by the Johnson algorithm.

1 234561	3 142536	5 231465	7 461352	9 314265
2 324651	4 361254	6 542631	8 624315	10 245361

Question 8

7 jobs must be executed on a machine M_1 then on a machine M_2 and finally on a machine M_3, in that order. At the end of the execution of a job on M_1, it can be executed on M_2 if the machine is available, otherwise, we wait for the availability to start the job on M_2. Similarly for M_3. The durations of the jobs on the three machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5	j_6	j_7
M_1	90	270	130	90	120	240	110
M_2	70	10	60	10	10	30	60
M_3	150	150	230	160	130	80	260

Parmi les ordres suivants, lequel correspond à un résultat que pourrait renvoyer l'algorithme de Johnson adapté au cas de 3 machines ? Si cet algorithme n'est pas adapté au cas de 3 machines, cochez la case "Err".
Check the order that could be returned by the Johnson algorithm, if this algorithm is suitable for this case. Otherwise, check "Err".

[1] Err	[3] 4173652	[5] 1723645	[7] 4537126	[9] 1347265
[2] 3251647	[4] 3612754	[6] 4517326	[8] 5327461	[10] 1372465

Question 9

8 jobs must be executed on a machine M_1 then on a machine M_2, in **any** order. A job can be executed on one machine at a time, and a machine can execute only one job at the same time. The durations of the jobs on the two machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5	j_6	j_7	j_8
M_1	140	870	230	20	250	50	20	160
M_2	280	1160	20	240	230	40	100	240

What is the minimum time we need to execute all the jobs on the machines ?

1 1940	3 1960	5 1830	7 1910	9 2030
2 2260	4 2310	6 2230	8 1670	10 1740

Question 10

5 jobs must be executed on a machine M_1 then on a machine M_2, in that order. At the end of the execution of a job on M_1, it can be executed on M_2 if the machine is available, otherwise, we wait for the availability to start the job on M_2. The durations of the jobs on the two machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5
M_1	50	20	150	50	190
M_2	160	130	90	20	110

Check the order that could be returned by the Johnson algorithm.

1 12435	3 53124	5 54321	7 21534	9 23145
2 51243	4 21543	6 21435	8 43251	10 12543

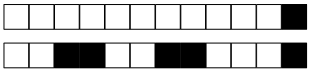
Question 11

8 jobs must be executed on a machine M_1 then on a machine M_2 and finally on a machine M_3, in that order. At the end of the execution of a job on M_1, it can be executed on M_2 if the machine is available, otherwise, we wait for the availability to start the job on M_2. Similarly for M_3. The durations of the jobs on the three machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5	j_6	j_7	j_8
M_1	140	320	210	140	190	300	160	180
M_2	80	80	70	30	20	40	10	30
M_3	170	150	80	90	150	240	250	320

Parmi les ordres suivants, lequel correspond à un résultat que pourrait renvoyer l'algorithme de Johnson adapté au cas de 3 machines ? Si cet algorithme n'est pas adapté au cas de 3 machines, cochez la case "Err".
Check the order that could be returned by the Johnson algorithm, if this algorithm is suitable for this case. Otherwise, check "Err".

[1] Err	[3] 67238514	[5] 61487235	[7] 52348176	[9] 64815327
[2] 81726354	[4] 64735182	[6] 46215378	[8] 78162534	[10] 42513687



Question 12

7 jobs must be executed on a machine M_1 then on a machine M_2, in **any** order. A job can be executed on one machine at a time, and a machine can execute only one job at the same time. The durations of the jobs on the two machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5	j_6	j_7
M_1	90	20	110	240	120	10	280
M_2	220	20	50	40	140	40	220

What is the minimum time we need to execute all the jobs on the machines ?

1 500	3 730	5 530	7 480	9 710
2 400	4 880	6 890	8 870	10 700

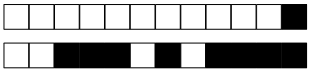
Question 13

7 jobs must be executed on a machine M_1 then on a machine M_2, in that order. At the end of the execution of a job on M_1, it can be executed on M_2 if the machine is available, otherwise, we wait for the availability to start the job on M_2. The durations of the jobs on the two machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5	j_6	j_7
M_1	130	200	80	40	20	110	150
M_2	230	50	50	60	80	150	270

Check the order that could be returned by the Johnson algorithm.

1 3174562	3 3521467	5 4276351	7 5427316	9 1452376
2 2467531	4 3712456	6 5461723	8 5473126	10 5136427



Question 14

6 jobs must be executed on a machine M_1 then on a machine M_2 and finally on a machine M_3, in that order. At the end of the execution of a job on M_1, it can be executed on M_2 if the machine is available, otherwise, we wait for the availability to start the job on M_2. Similarly for M_3. The durations of the jobs on the three machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5	j_6
M_1	100	140	210	200	60	70
M_2	50	10	60	20	20	60
M_3	120	130	110	80	80	170

Parmi les ordres suivants, lequel correspond à un résultat que pourrait renvoyer l'algorithme de Johnson adapté au cas de 3 machines ? Si cet algorithme n'est pas adapté au cas de 3 machines, cochez la case "Err".
Check the order that could be returned by the Johnson algorithm, if this algorithm is suitable for this case. Otherwise, check "Err".

1 Err	3 561324	5 621354	7 135264	9 654312
2 136452	4 452631	6 314256	8 452361	10 624513

Question 15

8 jobs must be executed on a machine M_1 then on a machine M_2, in **any** order. A job can be executed on one machine at a time, and a machine can execute only one job at the same time. The durations of the jobs on the two machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5	j_6	j_7	j_8
M_1	80	20	20	290	70	60	10	110
M_2	210	290	200	220	140	200	90	280

What is the minimum time we need to execute all the jobs on the machines ?

[1] 1440	[3] 1670	[5] 510	[7] 1290	[9] 1060
[2] 1630	[4] 470	[6] 1580	[8] 1300	[10] 660

Question 16

5 jobs must be executed on a machine M_1 then on a machine M_2, in that order. At the end of the execution of a job on M_1, it can be executed on M_2 if the machine is available, otherwise, we wait for the availability to start the job on M_2. The durations of the jobs on the two machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5
M_1	80	80	130	80	30
M_2	100	160	140	160	60

Check the order that could be returned by the Johnson algorithm.

1 12543	3 15423	5 34251	7 51423	9 32451
2 14523	4 23145	6 51243	8 42531	10 14253

Question 17

6 jobs must be executed on a machine M_1 then on a machine M_2 and finally on a machine M_3, in that order. At the end of the execution of a job on M_1, it can be executed on M_2 if the machine is available, otherwise, we wait for the availability to start the job on M_2. Similarly for M_3. The durations of the jobs on the three machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5	j_6
M_1	150	230	160	180	110	220
M_2	30	10	20	70	60	70
M_3	240	170	140	90	70	240

Parmi les ordres suivants, lequel correspond à un résultat que pourrait renvoyer l'algorithme de Johnson adapté au cas de 3 machines ? Si cet algorithme n'est pas adapté au cas de 3 machines, cochez la case "Err".
Check the order that could be returned by the Johnson algorithm, if this algorithm is suitable for this case. Otherwise, check "Err".

1 Err	3 154263	5 162345	7 634152	9 163254
2 134562	4 532146	6 514263	8 632145	10 263514

Question 18

7 jobs must be executed on a machine M_1 then on a machine M_2, in **any** order. A job can be executed on one machine at a time, and a machine can execute only one job at the same time. The durations of the jobs on the two machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5	j_6	j_7
M_1	230	120	160	150	100	170	180
M_2	200	80	160	120	180	260	280

What is the minimum time we need to execute all the jobs on the machines ?

[1] 530	[3] 700	[5] 1280	[7] 610	[9] 1240
[2] 800	[4] 460	[6] 1110	[8] 730	[10] 1060

Question 19

5 jobs must be executed on a machine M_1 then on a machine M_2, in that order. At the end of the execution of a job on M_1, it can be executed on M_2 if the machine is available, otherwise, we wait for the availability to start the job on M_2. The durations of the jobs on the two machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5
M_1	10	200	10	100	90
M_2	180	150	50	190	70

Check the order that could be returned by the Johnson algorithm.

1 51423	3 13425	5 41253	7 35412	9 52413
2 54123	4 12345	6 34215	8 21534	10 45321

Question 20

5 jobs must be executed on a machine M_1 then on a machine M_2 and finally on a machine M_3, in that order. At the end of the execution of a job on M_1, it can be executed on M_2 if the machine is available, otherwise, we wait for the availability to start the job on M_2. Similarly for M_3. The durations of the jobs on the three machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5
M_1	90	50	140	50	80
M_2	10	40	20	10	50
M_3	50	100	190	160	170

Parmi les ordres suivants, lequel correspond à un résultat que pourrait renvoyer l'algorithme de Johnson adapté au cas de 3 machines ? Si cet algorithme n'est pas adapté au cas de 3 machines, cochez la case "Err".
Check the order that could be returned by the Johnson algorithm, if this algorithm is suitable for this case. Otherwise, check "Err".

1 Err	3 41523	5 31425	7 25413	9 21354
2 34521	4 25143	6 32415	8 42531	10 12543

Question 21

7 jobs must be executed on a machine M_1 then on a machine M_2, in **any** order. A job can be executed on one machine at a time, and a machine can execute only one job at the same time. The durations of the jobs on the two machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5	j_6	j_7
M_1	250	140	210	40	140	30	210
M_2	80	280	10	250	230	130	160

What is the minimum time we need to execute all the jobs on the machines ?

[1] 790	[3] 420	[5] 1090	[7] 1200	[9] 1140
[2] 1210	[4] 1000	[6] 1020	[8] 330	[10] 670

Question 22

5 jobs must be executed on a machine M_1 then on a machine M_2, in that order. At the end of the execution of a job on M_1, it can be executed on M_2 if the machine is available, otherwise, we wait for the availability to start the job on M_2. The durations of the jobs on the two machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5
M_1	50	180	110	140	160
M_2	130	130	50	130	180

Check the order that could be returned by the Johnson algorithm.

1 54321	3 24315	5 43152	7 42153	9 15432
2 52134	4 32541	6 54231	8 15243	10 45312

Question 23

8 jobs must be executed on a machine M_1 then on a machine M_2 and finally on a machine M_3, in that order. At the end of the execution of a job on M_1, it can be executed on M_2 if the machine is available, otherwise, we wait for the availability to start the job on M_2. Similarly for M_3. The durations of the jobs on the three machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5	j_6	j_7	j_8
M_1	170	290	120	150	210	320	300	290
M_2	80	80	10	80	80	80	50	30
M_3	190	320	240	80	270	200	150	260

Parmi les ordres suivants, lequel correspond à un résultat que pourrait renvoyer l'algorithme de Johnson adapté au cas de 3 machines ? Si cet algorithme n'est pas adapté au cas de 3 machines, cochez la case "Err".
Check the order that could be returned by the Johnson algorithm, if this algorithm is suitable for this case. Otherwise, check "Err".

1 Err	3 23518746	5 31528674	7 18437625	9 38476521
2 67514823	4 12786435	6 18473652	8 75823164	10 64735218

Question 24

8 jobs must be executed on a machine M_1 then on a machine M_2, in **any** order. A job can be executed on one machine at a time, and a machine can execute only one job at the same time. The durations of the jobs on the two machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5	j_6	j_7	j_8
M_1	40	160	250	70	270	280	110	230
M_2	80	250	50	280	300	100	320	140

What is the minimum time we need to execute all the jobs on the machines ?

[1] 1350	[3] 1180	[5] 1410	[7] 870	[9] 570
[2] 1150	[4] 1490	[6] 1380	[8] 790	[10] 1520

Question 25

7 jobs must be executed on a machine M_1 then on a machine M_2, in that order. At the end of the execution of a job on M_1, it can be executed on M_2 if the machine is available, otherwise, we wait for the availability to start the job on M_2. The durations of the jobs on the two machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5	j_6	j_7
M_1	60	40	50	110	170	210	50
M_2	240	200	130	60	60	60	250

Check the order that could be returned by the Johnson algorithm.

1 1347526	3 7146532	5 2371456	7 4625713	9 2645713
2 6417253	4 2364517	6 6351427	8 5317426	10 4315627

Question 26

5 jobs must be executed on a machine M_1 then on a machine M_2 and finally on a machine M_3, in that order. At the end of the execution of a job on M_1, it can be executed on M_2 if the machine is available, otherwise, we wait for the availability to start the job on M_2. Similarly for M_3. The durations of the jobs on the three machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5
M_1	70	80	100	130	70
M_2	50	40	30	30	30
M_3	100	130	130	60	50

Parmi les ordres suivants, lequel correspond à un résultat que pourrait renvoyer l'algorithme de Johnson adapté au cas de 3 machines ? Si cet algorithme n'est pas adapté au cas de 3 machines, cochez la case "Err".
Check the order that could be returned by the Johnson algorithm, if this algorithm is suitable for this case. Otherwise, check "Err".

1 Err	3 43521	5 12345	7 42135	9 42513
2 35412	4 52143	6 51432	8 45213	10 32451

Question 27

7 jobs must be executed on a machine M_1 then on a machine M_2, in **any** order. A job can be executed on one machine at a time, and a machine can execute only one job at the same time. The durations of the jobs on the two machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5	j_6	j_7
M_1	160	140	190	170	1070	130	240
M_2	280	40	10	120	790	140	180

What is the minimum time we need to execute all the jobs on the machines ?

[1] 1690	[3] 2180	[5] 1790	[7] 2100	[9] 1660
[2] 1560	[4] 1650	[6] 2090	[8] 1670	[10] 1860

Question 28

7 jobs must be executed on a machine M_1 then on a machine M_2, in that order. At the end of the execution of a job on M_1, it can be executed on M_2 if the machine is available, otherwise, we wait for the availability to start the job on M_2. The durations of the jobs on the two machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5	j_6	j_7
M_1	230	20	180	230	220	50	90
M_2	240	150	100	70	180	140	270

Check the order that could be returned by the Johnson algorithm.

1 4375216	3 2435671	5 7645132	7 7231564	9 2546173
2 6247351	4 4725613	6 4521376	8 6431725	10 2671534

Question 29

7 jobs must be executed on a machine M_1 then on a machine M_2 and finally on a machine M_3, in that order. At the end of the execution of a job on M_1, it can be executed on M_2 if the machine is available, otherwise, we wait for the availability to start the job on M_2. Similarly for M_3. The durations of the jobs on the three machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5	j_6	j_7
M_1	240	120	210	230	240	180	210
M_2	80	30	60	60	30	60	30
M_3	200	260	190	120	100	170	80

Parmi les ordres suivants, lequel correspond à un résultat que pourrait renvoyer l'algorithme de Johnson adapté au cas de 3 machines ? Si cet algorithme n'est pas adapté au cas de 3 machines, cochez la case "Err".
Check the order that could be returned by the Johnson algorithm, if this algorithm is suitable for this case. Otherwise, check "Err".

1 Err	3 3457621	5 5423617	7 4576213	9 3416572
2 2136457	4 3674152	6 3157246	8 7354162	10 1645732

Question 30

6 jobs must be executed on a machine M_1 then on a machine M_2, in **any** order. A job can be executed on one machine at a time, and a machine can execute only one job at the same time. The durations of the jobs on the two machines are given in the following table.

Duration $t(j, M)$	j_1	j_2	j_3	j_4	j_5	j_6
M_1	40	780	150	240	240	100
M_2	10	700	100	240	230	100

What is the minimum time we need to execute all the jobs on the machines ?

1 1480	3 1640	5 1450	7 1340	9 1380
2 1370	4 1440	6 1490	8 1550	10 1600

Entraînement - Training

Noircissez complètement ci-dessous les 3 premières lettres de votre nom de famille et la première lettre de votre prénom. Par exemple, pour Jean Dupont, cochez J, D, U, P ; pour Henri Harley, cochez seulement H, A, R ; pour Bernard Ca, cochez seulement A, B, C.

Check entirely the 3 first letters of your lastname and the first letter of your firstname. For instance, for Jean Dupont, check J, D, U, P ; for Henri Harley, check only H, A, R ; for Bernard Ca, check only A, B, C.

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Then write your lastname and firstname below.

Nom et prénom :

.....

Les réponses aux questions sont à donner exclusivement sur cette feuille. Les réponses données sur les feuilles précédentes ne seront pas prises en compte. Pour cocher une case, il faut la **noircir complètement**. Vous pouvez effacer votre réponse à la gomme ou avec du blanc, attention à ne pas effacer la case à cocher. Si vous êtes dans l'impossibilité de corriger une erreur, cette page est dupliquée au verso ; vous pouvez alors barrer cette feuille ci et répondre au verso.

Question 1 :	1	2	3	4	5	6	7	8	9	10
Question 2 :	1	2	3	4	5	6	7	8	9	10
Question 3 :	1	2	3	4	5	6	7	8	9	10
Question 4 :	1	2	3	4	5	6	7	8	9	10
Question 5 :	1	2	3	4	5	6	7	8	9	10
Question 6 :	1	2	3	4	5	6	7	8	9	10
Question 7 :	1	2	3	4	5	6	7	8	9	10
Question 8 :	1	2	3	4	5	6	7	8	9	10
Question 9 :	1	2	3	4	5	6	7	8	9	10
Question 10 :	1	2	3	4	5	6	7	8	9	10
Question 11 :	1	2	3	4	5	6	7	8	9	10
Question 12 :	1	2	3	4	5	6	7	8	9	10
Question 13 :	1	2	3	4	5	6	7	8	9	10

Question 14 : 1 2 3 4 5 6 7 8 9 10

Question 15 : 1 2 3 4 5 6 7 8 9 10

Question 16 : 1 2 3 4 5 6 7 8 9 10

Question 17 : 1 2 3 4 5 6 7 8 9 10

Question 18 : 1 2 3 4 5 6 7 8 9 10

Question 19 : 1 2 3 4 5 6 7 8 9 10

Question 20 : 1 2 3 4 5 6 7 8 9 10

Question 21 : 1 2 3 4 5 6 7 8 9 10

Question 22 : 1 2 3 4 5 6 7 8 9 10

Question 23 : 1 2 3 4 5 6 7 8 9 10

Question 24 : 1 2 3 4 5 6 7 8 9 10

Question 25 : 1 2 3 4 5 6 7 8 9 10

Question 26 : 1 2 3 4 5 6 7 8 9 10

Question 27 : 1 2 3 4 5 6 7 8 9 10

Question 28 : 1 2 3 4 5 6 7 8 9 10

Question 29 : 1 2 3 4 5 6 7 8 9 10

Question 30 : 1 2 3 4 5 6 7 8 9 10